Лабораторна робота №5
Створення програмного коду ST - мовою

Знайомство з редактором ST – мови

Робочі листки тіла коду редагуються ST–мовою, використовуючи текстовий редактор. Інструкції та вирази користувач може друкувати або вставляти за допомогою Майстра редагування. Зручніше користуватися Майстром редагування тому, що він містить багато стандартних ключових слів, функцій та функціональних блоків, а це перешкоджає появі помилок при створені ST-програми. При друкуванні інструкцій та виразів рекомендується використовувати абзаци. Кожний рядок починається з номера, а кожна інструкція має закінчуватися крапкою з комою. При використанні Майстра редагування це здійснюється автоматично.

Коментарі створюються у круглих дужках із зірочками. Різні елементи синтаксису мають свій колір: ключові слова - блакитні, змінні та імена екземплярів блоків – чорні, коментарі – зелені.

Для редагування ST-мовою необхідно в проектному дереві відкрити робочий листок тіла коду, подвійно клацаючи по відповідній іконі теки Logical POUs(Логічні POUs).

Якщо Майстра редагування на екрані не видно, натисніть <SHIFT> + <F2> або <Alt>+<3> або в панелі інструментів клацніть по іконі Edit Wizard (Майстра редагування) .

Щоб зробити вставку інструкції необхідно:

– у робочому листку тіла коду визначити курсором місце, де нова інструкція має бути вставлена;

– відкрити вікно Group (Група) Майстра редагування і вибрати Keywords (Ключові слова), де приведено список ключових слів;

– вибрати зі списку бажану інструкцію, наприклад CASE, і двічі клацнути по ній лівою клавішею миші. Синтаксис інструкції автоматично вставиться в робоче поле тіла коду:

```
CASE (*EXPRESSION (must return an INT value*) OF
    (* VALUE  a*):   (*STATEMENTS*);(* VALUE can be a single value  *)
    (* VALUES b*):   (*STATEMENTS*);(* or a set of VALUES          *)
    (*      .    :        .      *) (* for Example:         *)
    (*      .    :        .      *) (*              1   : ....; *)
    (* VALUE  x*):   (*STATEMENTS*);(*              2..4: ....; *)
    ELSE (*STATEMENTS*);
END_CASE;
```

У наведеній мовній структурі замість фактичних змінних і їх значень приведені зеленим текстом коментарі, які оточені круглими дужками і зірочками.

При редагуванні коментарі замінюються фактичними операндами (змінними та значеннями).

Щоб декларувати нову змінну, надрукуйте її ім'я у бажаній позиції тіла коду і встановіть на неї курсор;

– в панелі інструментів клацніть по іконі Variable (Змінна), з'явиться діалогове вікно Variables (Змінні), де у першому рядку поля Name (Ім'я) показане ім'я позначеної змінної;

– виберіть відповідні уставки для змінній (Usage, Data Type та ін.) і у рядку Local Variable Groups (Група локальній змінній) активізуйте Default (За замовчуванням) і натисніть ОК. Змінна вставиться у робочий листок тіла коду, а її декларація автоматично увійде у робочий листок сітки змінних.

При редагування ST – тіла коду можна використовувати змінні, які вже декларовані і записані у список поля Name (Ім'я). Можна також замінити одну змінну на іншу, яка вже декларована в робочому листку сітки змінних.

Для того, щоб викликати функцію, використовують Edit Wizard (Майстер редагування) . Для цього:

– в робочому листку тіла коду курсором виберіть місце, куди нова функція має бути вставлена;

– відкрийте вікно Group (Група) Майстра редагування і виберіть Functions (Функції);

– клацніть лівою клавішею миші, майстер редагування покаже список функцій;

– виберіть бажану функцію і двічі клацніть по ній, функція автоматично вставиться у позицію, яку показує текстовий курсор;

– замініть зелені коментарі, що оточені круглими дужками і зірочками, необхідними елементами.

Щоб викликати функціональний блок, використовуючи Майстра редагування, необхідно у його вікні Group (Група) вибрати Functions blocks (Функціональні блоки) і подвійно клацнути лівою клавішею миші по бажаному функціональному блоці. У діалоговому вікні Variables (Змінні), що з'явиться, задекларувати його ім'я і натиснути ОК. Мовна структура функціонального блока автоматично вставиться у позначене місце робочого листка тіла коду, а його декларація увійде в групу локальних змінних робочого листка сітки змінних. Після цього, враховуючи підказки, що оточені круглими дужками і зірочками, необхідно надрукувати вхідні та вихідні змінні і константи.

Створення проекту користувача

Створити ST-мовою програму керування роботою двигуна згідно з алгоритмом наведеним у лабораторній роботі №2.

Для цього відкрийте діалог New Project (Новий проект), використовуючи пункт меню File (файл). Подвійно клацніть по Project Wizard (Майстер проекту) і пройдіть увесь шлях створення проекту аналогічно варіанту програмування LD-мовою, тобто призначте ім'я проекту та його POU, виберіть у даному випадку ST-мову програмування, бажані імена і типи конфігурації, ресурсу і задачі.

Коли новий проект з'явиться у вікні проектного дерева, можна починати програмування.

Програмування ST-мовою функціонального блока лічильника для підрахунку кількості натисків на кнопку старту двигуна.

На робочому полі проектного коду позначте курсором місце початку програмування і за допомогою Edit Wizard (Майстер проекту) у таблиці Group виберіть функціональний блок CTU;

- задекларуйте цей функціональний блок, як Motor-Count;
- клацніть ОК, діалогове вікно закриється, а на робочому листку проекту залишиться програма функціонального блока лічильника Motor_Count створеного ST- мовою:

-

```
Motor_Count(CU:=(* BOOL *),RESET:=(* BOOL *),PV:=(* INT *));
(* BOOL *):=Motor_Count.Q;
(* INT *):=Motor_Count.CV;
```

Для оголошення змінних блока Motor_Count:

- установіть курсор перед коментарем CU- входа лічильника і в панелі інструментів лівою клавішею миші клацніть по іконі Variable (Змінна);
- у діалоговому вікні Variable Properties (Властивості змінної), що з'явиться, у рядку Name надрукуйте ім'я змінної Motor_Start, за допомогою якої буде активізуватися двигун;
- заповнити інші рядки, з урахуванням того, що Motor_Start є зовнішньою локальною, дискретною змінною, фізична адреса якої %IX0.0;
- натисніть ОК, діалогове вікно закриється, а CU- входу лічильника Motor_Count присвоюється змінна Motor_Start.

По черзі установіть курсор перед коментарями RESET-входу, PV-входу, Q-виходу, CV- виходу лічильника і оголосіть їх змінні та константу, відповідно, Motor, INT#3, ще раз Motor з фізичною адресою %QX0.0 і Pressed;

Програмування непередбаченої зупинки двигуна:

- позначте курсором новий рядок для продовження програми;
- в панелі інструментів активізуйте ікону Edit Wizard (Майстер проекту) і у таблиці Group, що з'явиться на екрані, виберіть функціональний блок RS, який має властивості перемикача із домінантою вимикача, і надайте йому ім'я RS_1;

```
RS_1(SET:=(* BOOL *),RESET1:=(* BOOL *));
(* BOOL *):=RS_1.Q1;
```

Щоб оголосити змінні блока RS:

- установіть курсор перед коментарем SET- входу функціонального блока RS і оголосіть внутрішню дискретну змінну Out.

Оскільки двигун має зупинятися після закінчення установленого часу роботи, а також примусово у будь-який момент, для RESET1-входу треба оголосити дві альтернативні змінні. Тоді:

- установіть курсор перед коментарем RESET1-входу і в панелі інструментів клацніть по іконі Variable (Змінна);
- оголосіть першу альтернативну змінну Emergency_Stop (Непередбачена_Зупинка) з фізичною адресою %IX0.1;

- натисніть ОК, діалогове вікно закриється, а RESET1-входу присвоюється перша альтернативна змінна Emergency_Stop;

- надрукуйте поряд з нею логічний оператор OR і в панелі інструментів клацніть по іконі Variable (Змінна);

- у діалоговому вікні Variable Properties (Властивості змінної), що з'явиться, оголосіть ім'я другої альтернативної змінної Mot_Time, яка зупинить двигун за командою таймера;

- натисніть ОК і друга альтернативна змінна Mot_Time присвоюється RESET1-входу;

- установіть курсор перед коментарем змінної, яка присвоюється виходу Q1 функціонального блока RS і задекларуйте змінну Motor з адресою вихідного каналу модуля виводу %QX0.0;

Програмування часу роботи двигуна.

Позначте курсором новий рядок для продовження програми;

- в панелі інструментів активізуйте ікону Edit Wizard (Майстер проекту), у таблиці Group, що з'явиться на екрані, виберіть функціональний блок таймера TON і присвойте йому ім'я M_Time;

```
M_Time(IN:=(* BOOL *),PT:=(* TIME *));
(* BOOL *):=M_Time.Q;
(* TIME *):=M_Time.ET;
```

- оголосіть змінні:

для IN- входу Motor, для PT- входу константу T#20s, для Q- виходу Mot_Time і для ET- виходу Actual_Time.

Програмування лічильника активізацій двигуна:

- Позначте курсором новий рядок для продовження програми;

- активізуйте в панелі інструментів ікону Edit Wizard (Майстер проекту) і у таблиці Group з меню Keywords виберіть оператор IF:

```
IF  (*EXPRESSION (must return a boolean value)*)
    THEN        (*If returned value of EXPRESSION = TRUE*)
    (*STATEMENT*);
END_IF;
```

- установіть курсор після оператора IF і в панелі інструментів лівою клавішею миші клацніть по іконі Variable (Змінна);

- у діалоговому вікні Variable Properties (Властивості змінної), що з'явиться, зі списку Name виберіть вже оголошену змінну Out і натисніть ОК;

- установіть курсор після оператора THEN і надрукуйте вираз Cycle_Count:=Cycle_Count+1;

Для компіляції проекту в панелі інструментів натисніть ікону Make (Створювати).

Якщо помилки є, відкоригуйте проект, якщо немає - завантажте його, використовуючи відомий шлях: Project Control Dialog → Download→Download.

Натисніть кнопку Cold (Холодний) у діалоговому вікні Resource (Ресурс) для холодного запуску симулятора ПЛК;

- в панелі інструментів клацніть по іконі Debug on/off (Налагодження В/В), щоб перейти в оперативний режим роботи. При цьому усі змінні позначаться різними кольорами;
- клацніть лівою клавішею миші по Demoio_Driver, що унизу екрана, для відкриття симулятора I/O;
- тричі подвійно клацніть лівою клавішею миші по нульовому світлодіоду нульового модуля вводу In.

Програма почне виконуватися, а біля змінної Actual_Time таймера M_Time будуть змінюватися секунди затримки. При цьому з кожним новим запуском програми вміст лічильника Cycle_Count збільшується на одиницю.

```
        (*Програмування лічильника кількості натисків стартової
         кнопки*)
FALSE   Motor_Count(CU:=Motor_Start,RESET:=Motor,PV:=INT#3);
FALSE   out:=Motor_Count.Q;
    0   Pressed(* INT *):=Motor_Count.CV;
        (*Програмування непередбаченої зупинки двигуна*)
FALSE   RS_1(SET:=out,RESET1:=Emergency_Stop OR Mot_Time);
 TRUE   Motor:=RS_1.Q1;
        (*Програмування таймера часу роботи двигуна*)
 TRUE   M_Time(IN:=Motor,PT:=T#20s);
FALSE   Mot_Time:=M_Time.Q;
5.700   Actual_Time:=M_Time.ET;
        (*Програмування лічильника активацій двигуна*)
FALSE   IF  out
    2       THEN Cycle_Count:=Cycle_Count+1;
        END_IF;
```

Завдання для самостійної роботи

1. Розробити програму керування роботою двох двигунів, які після запуску одного з двигунів безперервно по черзі вмикаються і вимикаються. Тривалість роботи двигунів відповідно 5 і 10 секунд. Початковий запуск першого двигуна здійснюється одноразовим натисненням пускової кнопки, а початковий запуск другого двигуна – дворазовим. При цьому двигуни можна у будь-яку мить зупинити, а кожне вмикання двигунів підраховується лічильником. Коли кількість вмикань двигунів досягає десяти, їх робота автоматично зупиняється.

2. Розробити програму зміни рівня речовини при керуванні ним в автоматичному та ручному режимах. В ручному режимі рівень змінюється під впливом кнопок «Вверх» і «Униз» в діапазоні 0-100%, а в автоматичному – за рахунок позиційного регулювання в діапазоні 40-100%.

Контрольні запитання

1. Як відкрити робочий листок тіла коду для редагування ST–мовою?
2. Як декларувати нову змінну?
3. Як можна використати вже декларовану змінну при редагуванні тіла коду?
4. Як викликаються функції та функціональні блоки?

5. Як запрограмувати ST-мовою лічильник активізацій двигуна?
6. Як оголосити змінні функціонального блока TON?
7. Навіщо у програмі використовується функціональний блок RS?
8. Як здійснюється налагодження програми?